

[illegible]

Syn
--
LII
LII
LII
LII
LII
LII
LII
LII
LII
LII
LII
LII
LII
LII
LII
MEI
MEI
RMI
RMI
SUI
SUI
SUI
SUI
SUI
SUI
SUI
SUI
SUI
SUI
SUI
SUI
SUI
SUI
SUI
SYN
SYN
SYN
SYN
SYN
SYN

```

LL               IIIIII             SSSSSSSS
LL               IIIIII             SSSSSSSS
LL               II                 SS
LL               II                 SS
LL               II                 SS
LL               II                 SS
LL               II                 SSSSSS
LL               II                 SSSSSS
LL               II                 SS
LL               II                 SS
LL               II                 SS
LL               II                 SS
LL               II                 SSSSSSSS
LLLLLLLLLLLLLL IIIIII             SSSSSSSS
LLLLLLLLLLLLLL IIIIII             SSSSSSSS

```


(2)	52	DECLARATIONS
(3)	81	SUM\$OPEN
(4)	187	RETURN_FNS

06

```
0000 1      .TITLE  SUM$OPEN
0000 2
0000 3      .IDENT  'V04-000'
0000 4
0000 5
0000 6 *****
0000 7 *
0000 8 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 *  ALL RIGHTS RESERVED.
0000 11 *
0000 12 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 *  TRANSFERRED.
0000 18 *
0000 19 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 *  CORPORATION.
0000 22 *
0000 23 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *
0000 27 *****
0000 28
0000 29
0000 30 ++
0000 31 : FACILITY:
0000 32
0000 33 : ABSTRACT:
0000 34
0000 35 :      Source Update Merge procedure to check the existance of
0000 36 :      the files in an update files list.
0000 37
0000 38
0000 39
0000 40 : ENVIRONMENT:  USER MODE
0000 41
0000 42 : AUTHOR:      R. Newland
0000 43
0000 44
0000 45 : MODIFIED BY:
0000 46
0000 47 :      V02-001      BLS0139      Benn Schreiber      26-Jan-1981
0000 48 :      Remove deallocation of unused part of resultant name buffer
0000 49
0000 50 :--
```

DECLARATIONS

```
0000 52      .SBTTL  DECLARATIONS
0000 53      :
0000 54      :
0000 55      : Macro definitions
0000 56      :
0000 57      DEFUPFBK      ; Source update merge
0000 58      $FABDEF      ; FAB block
0000 59      $NAMDEF      ; NAM block
0000 60      :
0000 61      :
0000 62      :
0000 63      :
0000 64      : Local storage
0000 65      :
0000 66      :
00000000 67      .PSECT  SUM$RO_DATA,NOEXE,NOWRT,LONG
0000 68      :
0000 69      :
0000 70      SUM_DEFNAME:
44 50 55 2E 00' 0000 71      .ASCIC  /.UPD/      ; Default file name
04 0000
0005 72      :
0005 73      SUM_MAXRSS:
00000100 0005 74      .LONG  256      ; Max size of resultant file name string
0009 75      :
0009 76      :
0009 77      :
00000000 78      .PSECT  SUM$RW_DATA,NOEXE,LONG
0000 79      ;
```


SUM\$OPEN

```
0000 81 .SBTTL SUM$OPEN
0000 82 :
0000 83 :++
0000 84 : Functional description:
0000 85 :
0000 86 : This procedure determines which files in an update files list
0000 87 : can be successfully opened. The NAM block information of each
0000 88 : file which can be opened is saved so that subsequent opens can
0000 89 : be performed quickly. The resultant file specification string
0000 90 : is also saved. For files that can not be opened because of a
0000 91 : 'file not found' error a warning message is issued and the file
0000 92 : entry removed. For files that can not be opened for any other
0000 93 : reason a fatal error message is issued.
0000 94 :
0000 95 : The NAM block of the associated source file is used to
0000 96 : supply initial default information.
0000 97 :
0000 98 : The input parameter, address of first update file, will be
0000 99 : changed if the first file (or the new first file) can not be
0000 100 : found. If all files can not be found the pointer will become zero.
0000 101 :
0000 102 :
0000 103 : Calling sequence:
0000 104 :
0000 105 : CALLG/CALLS
0000 106 :
0000 107 :
0000 108 : Inputs:
0000 109 :
0000 110 : 4(AP) = Source file NAM block address
0000 111 : 8(AP) = Pointer to address of first entry in Update list
0000 112 :
0000 113 : Outputs:
0000 114 :
0000 115 : None
0000 116 :
0000 117 :
0000 118 :--
0000 119 :
0000 120 : .PSECT SUM$CODE,NOWRT, LONG
0000 121 :
0000 122 : .DEBUG SUM$OPEN
0000 123 :
0000 124 : .ENTRY SUM$OPEN, ^M<R2,R3,R4,R10,R11>
0002 125 :
0002 126 : MOVAB SUM$FAB,R2 ; Set FAB pointer
35 52 00000000'EF 9E 0002 127 : MOVAB SUM_DEFNAME,FAB$B_DNS(R2) ; Initialise default file name
30 A2 00000000'EF 90 0009 128 : MOVAL SUM_DEFNAME+1,FAB$B_DNA(R2)
00 04 A2 18 E5 0011 129 : BBCC #FAB$V_NAM,FAB$B_FOP(R2),10$ ; Set for open by file name
001E 130 10$:
001E 131 : MOVL 4(AP),R3 ; Set related NAM block pointer
5A 08 BC D0 0022 132 : MOVL @8(AP),R10 ; and Update file block pointer
5B 08 BC DE 0026 133 : MOVAL @8(AP),R11 ; Set pointer to previous block
002A 134 20$:
002A 135 : MOVAB UPF_T_NAM(R10),R4 ; Set NAM block pointer
34 A2 20 AA 9E 002E 136 : MOVAB UPF_Q_CMNT(R10), - ; Set file name string size and address
0033 137 : FAB$B_FNS(R2)
```


2C A2	24 AA	D0	0033	138	MOVL	UPF_Q_CMNT+4(R10), -	
			0038	139		FAB\$FNA(R2)	
28 A2	54	D0	0038	140	MOVL	R4,FAB\$FNA(R2)	; Set NAM block pointer
10 A4	53	D0	003C	141	MOVL	R3,NAM\$RLF(R4)	; Set related NAM block pointer
0A A4	FF 8F	90	0040	142	MOVB	#255,NAM\$B_ESS(R4)	; Set size of expanded string
02 A4	FF 8F	90	0045	143	MOVB	#255,NAM\$B_RSS(R4)	; Set size of resultant string buffer
	04 A4	DF	004A	144	PUSHAL	NAM\$RSA(R4)	; Address to store resultant string
00C00005	'EF	DF	004D	145	PUSHAL	SUM_M\$XRSS	; Address of max resultant string length
00000000	'GF	02	FB	0053	CALLS	#2,G^LIB\$GET_VM	; Get memory block for resultant string
	6E 50	E9	005A	147	BLBC	R0,50\$; Error if LBC
0C A4	04 A4	D0	005D	148	MOVL	NAM\$RSA(R4),NAM\$ESA(R4)	; Set expanded string buffer
			0062	149	\$OPEN	FAB=R2,ERR=SUM\$OPEN_ERR	; Open file
	3F 50	E8	0071	150	BLBS	R0,40\$; OK if LBS
000000C0	'8F	D1	0074	151	CMPL	R0,#RM\$FNF	; Was error 'file not found'?
	03	13	007B	152	BEQL	30\$; Yes if EQL
	0060	31	007D	153	BRW	90\$	
			0080	154			30\$:
	04 A4	DF	0080	155	PUSHAL	NAM\$RSA(R4)	; Return resultant file spec block
00000005	'EF	DF	0083	156	PUSHAL	SUM_M\$XRSS	
00000000	'GF	02	FB	0089	CALLS	#2,G^LIB\$FREE_VM	
	46 50	E9	0090	158	BLBC	R0,80\$; Error if LBC
	004B	30	0093	159	BSBW	RETURN_FNS	; Return file name string block
	40 50	E9	0096	160	BLBC	R0,80\$; Error if LBC
	6R	DF	0099	161	PUSHAL	(R11)	; Return the update block
00000000	'EF	DF	009B	162	PUSHAL	SUM\$BLSIZE	; Size of block
	5A 6A	D0	00A1	163	MOVL	(R10),R10	; Advance pointer before returning block
00000000	'GF	02	FB	00A4	CALLS	#2,G^LIB\$FREE_VM	; Free memory
	2B 50	E9	00AB	165	BLBC	R0,80\$; Error if LBC
	6B 5A	D0	00AE	166	MOVL	R10,(R11)	; Relink update blocks
	21	11	00B1	167	BRB	70\$	
			00B3	168			40\$:
			00B3	169	\$CLOSE	FAB=R2,ERR=SUM\$CLOSE_ERR	; Close file
	1B 50	E9	00C2	170	BLBC	R0,90\$; Error if LBC
53	54	D0	00C5	171	MOVL	R4,R3	; Next related NAM block will be
			00C8	172			; this files' NAM block
	0016	30	00C8	173	BSBW	RETURN_FNS	; Return file name string block
			00CB	174			50\$:
	0B 50	E9	00CB	175	BLBC	R0,80\$; Error if LBC
			00CE	176			:
5B	5A	D0	00CE	177	MOVL	R10,R11	; Form previous block pointer
5A	6A	D0	00D1	178	MOVL	(R10),R10	; Point to next Update file block
			00D4	179			70\$:
	0A	13	00D4	180	BEQL	90\$; and finish if end of list
	FF51	31	00D6	181	BRW	20\$; Go back for next file
			00D9	182			80\$:
00000000	'EF	00	FB	00D9	CALLS	#0,SUM\$LIB_ERR	; Report library error
			00E0	184			90\$:
		04	00E0	185	RET		

RETURN_FNS

```
00E1 187      .SBTTL RETURN_FNS
00E1 188      :
00E1 189      :
00E1 190      Subroutine to return file name string block to virtual memory
00E1 191      :
00E1 192      Inputs:
00E1 193      :
00E1 194      R10 = Address of SUM block
00E1 195      :
00E1 196      Outputs:
00E1 197      :
00E1 198      R0 = Success/error status
00E1 199      :
00E1 200      :
00E1 201      RETURN_FNS:
20 AA B5 00E1 202      TSTW UPF_Q_CMNT(R10)      ; Was there a file name string?
      OD 13 00E4 203      BEQL 10$      ; No if EQL so no memory block to return
24 AA DF 00E6 204      PUSHAL UPF_Q_CMNT+4(R10) ; Address of file name string
20 AA DF 00E9 205      PUSHAL UPF_Q_CMNT(R10) ; Size of string
00000000'GF 02 FB 00EC 206      CALLS #2,G^CIB$FREE_VM ; Return memory
      00F3 207 10$:
05 00F3 208      RSB
      00F4 209      :
      00F4 210      :
      00F4 211      .END
```


SUM\$OPEN
Symbol table

D 10

16-SEP-1984 02:12:19 VAX/VMS Macro V04-00
5-SEP-1984 03:39:19 [SUM.SRC]SUMOPEN.MAR;1

Page 6
(4)

\$\$TMP1	=	00000002		
\$\$TMP2	=	00000052		
BIT...	=	00000005		
FAB\$B_DNS	=	00000035		
FAB\$B_FNS	=	00000034		
FAB\$B_DNA	=	00000030		
FAB\$B_FNA	=	0000002C		
FAB\$B_FOP	=	00000004		
FAB\$B_NAM	=	00000028		
FAB\$V_NAM	=	00000018		
LIB\$FREE_VM	*****		X	04
LIB\$GET_VM	*****		X	04
NAM\$B_ESS	=	0000000A		
NAM\$B_RSS	=	00000002		
NAM\$K_BLN	=	00000060		
NAM\$B_ESA	=	0000000C		
NAM\$B_RLF	=	00000010		
NAM\$B_RSA	=	00000004		
RETURN_FNS	000000E1		R	04
RMSS_FNF	*****		X	04
SIZ...	=	00000001		
SUM\$BLSZE	*****		X	04
SUM\$CLOSE_ERR	*****		X	04
SUM\$FAB	*****		X	04
SUM\$LIB_ERR	*****		X	04
SUM\$OPEN	00000000		RG D	04
SUM\$OPEN_ERR	*****		X	04
SUM_DEFNAME	00000000		R	02
SUM_MAXRSS	00000005		R	02
SUM_M_AUDIT	=	00000001		
SUM_M_AUDITNEW	=	00000002		
SUM_M_DELETE	=	00000010		
SUM_M_SRCUPD	=	00000004		
SUM_M_SUBCLSH	=	00000008		
SUM_V_AUDIT	=	00000000		
SUM_V_AUDITNEW	=	00000001		
SUM_V_DELETE	=	00000004		
SUM_V_SRCUPD	=	00000002		
SUM_V_SUBCLSH	=	00000003		
SYSS\$CLOSE	*****		GX	04
SYSS\$OPEN	*****		GX	04
UPF_B_EDFLAGS	00000009			
UPF_B_FIFLAGS	00000008			
UPF_B_FILENO	0000000C			
UPF_K_BLN	00000098			
UPF_L_PTR	00000000			
UPF_Q_AUDDS	00000018			
UPF_Q_CMNT	00000020			
UPF_Q_EDITS	00000010			
UPF_T_AUDST	00000028			
UPF_T_NAM	00000038			
UPF_W_DOT	0000000A			
UPF_W_LOC1	00000004			
UPF_W_LOC2	00000006			

UPC
V04

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000098 (152.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SUM\$RO_DATA	00000009 (9.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
SUM\$RW_DATA	00000000 (0.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SUM\$CODE	000000F4 (244.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.31
Command processing	105	00:00:00.53	00:00:02.22
Pass 1	194	00:00:04.29	00:00:09.68
Symbol table sort	0	00:00:00.40	00:00:00.89
Pass 2	55	00:00:00.87	00:00:02.16
Symbol table output	7	00:00:00.07	00:00:00.25
Psect synopsis output	2	00:00:00.03	00:00:00.11
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	394	00:00:06.29	00:00:15.63

The working set limit was 1200 pages.
21065 bytes (42 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 379 non-local and 9 local symbols.
211 source lines were read in Pass 1, producing 20 object records in Pass 2.
17 pages of virtual memory were used to define 15 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SUM.OBJ]SUM.MLB;1	2
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	11

492 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SUMOPEN/OBJ=OBJ\$:SUMOPEN MSRC\$:SUMOPEN/UPDATE=(ENH\$:SUMOPEN)+LIB\$:SUM/LIB

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY